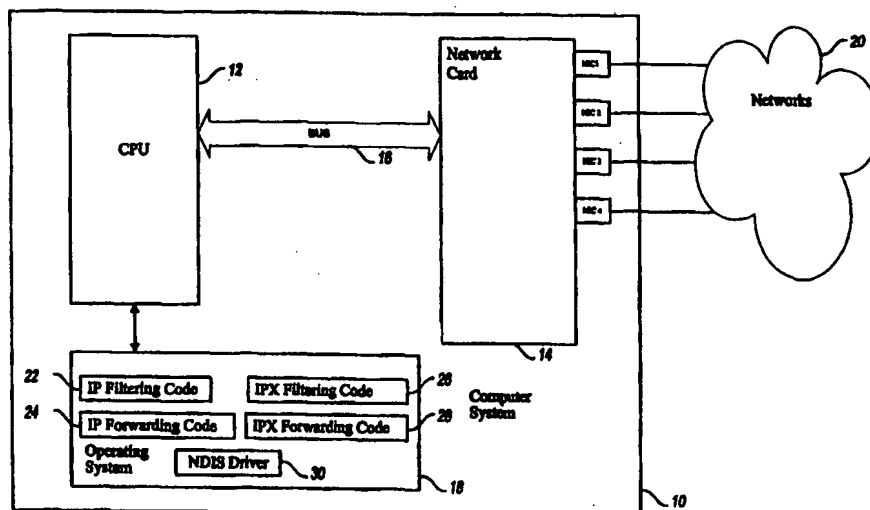




INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁶ : G06F 15/00	A2	(11) International Publication Number: WO 99/03044 (43) International Publication Date: 21 January 1999 (21.01.99)
(21) International Application Number: PCT/US98/14135 (22) International Filing Date: 8 July 1998 (08.07.98) (30) Priority Data: 08/891,020 10 July 1997 (10.07.97) US (71) Applicant: MICROSOFT CORPORATION [US/US]; One Microsoft Way, Redmont, WA 98052-6399 (US). (72) Inventors: KHAKI, Mohamed, J.; 1812 223rd Place N.E., Redmond, WA 98053 (US). PALL, Gurdeep, Singh; 220 Waverly Way, Kirkland, WA 98033 (US). (74) Agents: PIRIO, Maurice, J. et al.; Seed & Berry LLP, 6300 Columbia Center, 701 Fifth Avenue, Seattle, WA 98104-7092 (US).		(81) Designated States: JP, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE). Published <i>Without international search report and to be republished upon receipt of that report.</i>

(54) Title: FAST-FORWARDING AND FILTERING OF NETWORK PACKETS IN A COMPUTER SYSTEM**(57) Abstract**

A computer system facilitates concurrent filtering and fast-forwarding of network packets. The computer system includes a network card for interfacing with multiple computer networks, and the network adapter is able to fast-forward network packets that it receives to other network destinations without assistance from the main central processing in the computer system. A fast-forwarding cache is stored on the network card to facilitate such fast-forwarding. Network packets that are not fast-forwarded are processed by the main central processing of the computer system that may apply filtering to the network packets and route network packets accordingly. The main central processing unit provides routing information to the network card so that subsequently received network packets can be transmitted to the network destinations without intervention by the main central processing unit.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav	TM	Turkmenistan
BF	Burkina Faso	GR	Greece		Republic of Macedonia	TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's	NZ	New Zealand		
CM	Cameroon		Republic of Korea	PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakhstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

FAST-FORWARDING AND FILTERING OF NETWORK PACKETS IN A COMPUTER SYSTEM

TECHNICAL FIELD

The present invention relates generally to data processing systems
5 and more particularly to routing and filtering of network packets in a networked
computer environment.

BACKGROUND OF THE INVENTION

Computer systems are often interconnected into vast computer
networks. The computer systems connected on such networks communicate
10 with each other by sending information through their electronic connections.
The networks can be organized into various types of topologies. Figure 1
illustrates one such topology. A network 100 comprises several local area
networks 101-102 that are connected to a routing system 103. The computer
systems of each local area network are connected to the communications link
15 101a-102a. When a source computer system on a local area network sends
information to a destination computer system on the same local area network, the
source computer system prepares a packet that includes the address of the
destination computer system and transmits the packet on the communications
link. The other computer systems on that same local area network (*i.e.*,
20 connected to the communications link) read the packet that was transmitted. The
destination computer system detects that its address is included in that packet,
and it processes the packet accordingly. Because of geographic and speed
considerations, local area networks typically only include a limited number of
computer systems that are in close proximity. For example, a company with
25 offices in several locations may have a local area network at each location.
However, the users of the computer systems may need to send packets to one

another regardless to which of local area networks the users' computer systems are connected.

To allow packets to be sent from one local area network to another local area network, routing systems have been developed. A routing system is typically a dedicated special-purpose computer system to which each local area network is connected. The routing system maintains a cross-reference between computer system addresses and the local area network to which each computer system is connected. The routing system monitors the packets sent on each local area network to detect (using the cross-reference) when a computer system on one local area network is sending a packet to a computer system on another local area network. When the routing system detects such a packet, it forwards that packet onto the communications link for the local area network to which the destination computer system is connected. In this way, the routing system interconnects each of the local area networks into an overall network. Similar routing techniques are used to interconnect networks other than local area networks. For example, such routing techniques can be used on wide area networks and on the Internet.

Many different protocols have been developed to allow two computer systems to exchange information. If two computer systems support the same protocol, then they can exchange information. Certain protocols have been tailored to support the exchange of certain types of information efficiently. For example, the Internet protocol ("IP") was specified by the Department of Defense to facilitate the exchange of information between geographically separated computer systems. The IP specifies a destination in a packet format that identifies source and destination computer systems for data to exchange, but does not specify the format of the data itself. Several additional protocols may be used in conjunction with the IP to specify the format of the data. Once such additional protocol is the transmission control protocol ("TCP"). The TCP

further specifies sub-protocols, such as the hyper-text transmission protocol ("HTTP") and the file transfer protocol ("FTP"), which specify the format of the data of the packet.

Figure 2 is a diagram illustrating a typical packet sent on a local
5 area network. The packet includes a network routing header followed by
protocol specific data. The network routing header may include the destination
computer address, the source computer address, and the length of the packet.
The protocol specific data includes identification of the protocol and in the case
of the IP includes the IP destination address, the IP source address, and the
10 length of the IP portion of the packet. The data portion of the packet contains the
sub-protocol identification plus other data of the packet.

Conventional routing systems often use a caching mechanism to
speed the routing of the packets. When a packet is received, the routing system
accesses its cross-reference of the local area networks and determines to which
15 local area network the packet is to be routed. The routing system then stores that
routing information in a cache. When subsequent packets are received, the
routing system checks its cache and if the routing information for the packet is in
the cache, then the routing system forwards the packet in accordance with the
information in the cache. Although such a caching technique can help speed the
20 routing of packets, such caching techniques are not used by routing systems
when packet filtering is applied.

Packet filtering is often applied by a network administrator to
restrict the type and timing of packets that are sent over the network. For
example, a network administrator may want to restrict packets that are generated
25 by a computer game from being transmitted over the network during normal
business hours. A packet for a computer game may be identifiable, for example,
by a TCP destination address, that indicates which application on the computer
system identified by the IP destination address that is to receive the packet.

Thus, the network administrator would configure the routing system to not forward any such packets during normal business hours. Also, the network administrator may want to filter out packets based on their source and destination addresses. For example, a company CEO may only want to receive packets from
5 certain source computer systems and not every computer system on the network. When filtering is applied in conventional routing systems, caching is disabled and the slower procedure of checking the cross-reference information and filtering information is used for every packet. Thus, the processing not only of such filtered packets is slowed, but also of all packets is slowed because the
10 cache is disabled.

SUMMARY OF THE INVENTION

In accordance with a first aspect of the present invention, a method of fast-forwarding a network packet is performed in a general-purpose computer system. "Fast-forwarding" refers to the network card performing the routing
15 rather than a main central processing unit performing the routing. The computer system has a main central processing unit and a network card for interfacing the computer system with multiple networks. The network packet is received in the network card and is destined to a selected one of the networks. The received network packet is analyzed by the network card to determine whether the
20 network packet should be fast-forwarded to its destination network by the network card or alternatively, routed by the main central processing unit. When it is determined that a network packet should be fast-forwarded to the destination network by the network card, it is fast-forwarded without intervention of the main central processing unit.

25 In accordance with another aspect of the present invention, a network card is provided for interfacing a computer system with multiple networks. The network card includes a fast-forwarding mechanism and multiple

ports for interfacing with the multiple networks. The fast-forwarding mechanism fast-forwards a network packet that is received in the network card to a destination network without intervention by a main central processing unit of the computer system.

5 In accordance with a further aspect of the present invention, fast-forwarding and filtering may be concurrently active. In particular, a method for processing network packets that are received at a network card is practiced on a computer system. Fast-forwarding is active so that selected network packets that are received in the network card are fast-forwarded to at least one destination
10 network by the network card without main central processing unit intervention. Filtering is concurrently active with the fast-forwarding so that the main central processing unit applies filtering to at least some of the network packets received in the network card to determine whether to accept or drop the network packets.

BRIEF DESCRIPTION OF THE DRAWINGS

15 Figure 1 illustrates one topology of interconnected networks.

Figure 2 is a diagram illustrating a typical packet sent on a local area network.

Figure 3 is a block diagram of a general-purpose computer system 10 upon which the fast-forward system executes.

20 Figure 4 is a block diagram that illustrates components of the network card in more detail.

Figures 5A-5D illustrate the contents of the fast-forwarding cache as packets are received and sent.

Figures 6A-6E depict the fields in a number of different types of
25 packets.

Figure 7 is a flow diagram of the routine of the network card that receives a packet from the network.

Figure 8 is a flow diagram of a routine for the network card to receive a packet from the NDIS driver.

Figure 9 is a flow diagram of a routine of the operating systems to process a packet that is received from the NDIS driver.

5 Figure 10 is a flow diagram of an operating system routine to update the routing tables.

DETAILED DESCRIPTION OF THE INVENTION

The present invention provides a method and system for routing packets using a caching mechanism even when filtering is to be applied to some
10 of the packets. The routing system of the present invention, also referred to as the fast-forwarding system, is implemented on a general-purpose computer system in a way that allows caching of routing information and in a way that minimizes the overhead placed on the main central processing unit of the computer system. Thus, the fast-forwarding system not only allows for caching
15 when filtering, but also allows a general-purpose computer system to employ a caching mechanism.

The fast-forwarding system is described below relative to an embodiment that forwards and filters Internet protocol (IP) packets as well as Internet protocol extended (IPX) packets. Nevertheless, those skilled in the art
20 will appreciate that the present invention is not limited to embodiments that operate solely with these types of network packets; rather, the present invention may be also practiced with other types of network packets.

Figure 3 is a block diagram of a general-purpose computer system 10 upon which the fast-forward system executes. The computer system
25 is connected to a number of different computer networks 20. The computer system includes a central processing unit (CPU) 12 that is connected to a network card 14 via a bus 16. The network card includes multiple ports that are

referred to as network interface connections ("NICs"). Each NIC is connected to a communications link of a network and can be used to send and receive packets on the network. Typical types of network cards include 802.3, DIX, TR, Fast Ethernet, and FDDI cards. The computer system runs an operating system 18, such as a network operating system. For purposes of the discussion below, it is assumed that the operating system is the Microsoft® Windows® NT network operating system sold by Microsoft Corporation of Redmond, Washington. The operating system includes a number of different components: an IP filtering component 22 for performing filtering of IP network packets and an IP forwarding component 24 for routing IP network packets. The operating system also includes an IPX filtering component 26 and an IPX forwarding component 28 for routing IPX network packets. The operating system also has a network driver interface specification (NDIS) driver component 30. The NDIS driver component provides an interface between the operating system and the network card. The operating system invokes services of the NDIS driver to send and receive network packets.

Figure 4 is a block diagram that illustrates components of the network card in more detail. The network card includes components for interfacing the routing system with a number of different computer networks. The network card includes a number of NICs. These NICs are designated as NIC1, NIC2, NIC3, and NIC4. The network card also includes a microprocessor 32, an Ethernet chip 34, and storage 36, such as random access memory (RAM). The storage contains a fast-forwarding cache 76 and an incoming packet cache 80. Although one fast-forwarding cache and one incoming packet cache are shown, those skilled in the art will appreciate that separate versions of these caches may be used for IP network packets and IPX network packets. The fast-forwarding cache holds information that facilitates the fast-forwarding of network packets by the network card. As will be described in

more detail below, the fast-forwarding cache serves as a lookup table to which incoming network packets may be compared to determine whether they should be fast-forwarded by the network card.

In an alternative embodiment where multiple network cards are employed, the caches may be stored in shared memory on the computer system 10. When a network card receives a packet, it accesses the shared memory to access the fast-forwarding cache without interrupting the main central processing unit. Also, each network card has access to a mapping between the local area networks that are connected to each network card. In this way, each network card can route packets to local area networks connected to other network cards.

Figures 5A-5D illustrate the contents of the fast-forwarding cache as packets are received and sent. The cache contains a row for each unique packet identifier of the packets that are to be fast-forwarded. Each row contains a packet identifier that identifies packets to be fast-forwarded, a network address that indicates the destination computer system, a framing field that indicates the type of framing that is to be applied to the packet, and a NIC port number that identifies the port of the network card through which the packets with that packet identifier are to be transmitted. The packet identifier is a combination of fields from the packet that identifies packets that are to be fast-forwarded in the same manner. As shown by Figure 5A, the fast-forwarding cache is initially empty. When the network card receives a packet, it checks the cache to determine whether a row with the corresponding packet identifier is in the cache. In this case, since the cache is empty, there are no rows and the network card forwards the packet via the NDIS driver to the operating system. The network card also stores the packet in the incoming packet cache. When the operating system receives the packet, it determines whether any filtering applies and whether the packet needs to be forwarded onto another network. If no filtering applies and

the packet is to be forwarded, then the operating system returns the packet to the network card via the NDIS driver for forwarding on the appropriate local area network. When the network card receives the packet, it stores an entry in the fast-forwarding cache as shown in Figure 5B. In this case, the packet identifier

5 “<A5, B2, TCP, 14, 25>” indicates that the IP source address is “A5,” the IP destination is address “B2,” the protocol is “TCP,” the TCP source port is 14, and the TCP destination port is 25. The network address to which the packets with that package identifier are to be routed is “E7.” The outgoing packet is to be framed according to 802.3 protocol and the packets should be routed through

10 NIC port number 3. The network card then retrieves the packet from the incoming packet cache, prepares the packet for routing, and transmits the packet via the NIC port. When the network card receives the next packet with that packet identifier, it generates the packet identifier and determines whether a row with that packet identifier is in the fast-forward cache. If so, the network card,

15 retrieves the row and updates the network address for the packet, adjusts the framing according to the 802.3 protocol, and transmits the packet onto NIC port number 3. As part of processing, the network card also decrements the time-to-live flag of the packet. The time-to-live flag is initially set to the number of times the packet can be routed on its way to the destination. If the time-to-live

20 flag drops below zero, then the packet cannot be routed again and the network card drops the packet by not forwarding it. Also, if the checksum in the packet is incorrect, the network card also drops the packet.

Eventually, the network card will receive a packet with a different packet identification. The network card generates a packet identifier for that

25 packet and determines that there is no row in the cache for that packet identifier. The network card then forwards that packet to the operating system. The operating system applies the appropriate filtering. In this case, the operating system determines that the packet should not be forwarded. Thus, the operating

system does not forward the packet to the network card. The network card thus does not update the cache to contain a row for the packet identifier. Thus, when the next packet is received with that packet identifier, the network card generates the packet identifier and determines that there is no row in the cache for that
5 packet identifier and sends the packet to the operating system. Figure 5C illustrates the contents of the cache after a packet for another packet has been received and the fast-forwarding cache updated for that packet. In this case, the packet identifier is "<E1, B2, UDP, 36, 97>," the network cache address is "A5," the framing is Ethernet, and the NIC port address is 1. Occasionally, the
10 operating system may receive requests to change its routing information. For example, a computer system may be removed from one local area network and installed in another local area network. In such a case, packets routed to that computer system would need to be transmitted through a different NIC port. When the operating system updates its routing information, it also instructs the
15 network card to flush its fast-forwarding cache because the information in the cache may no longer be valid. The network card flushes its cache by removing all entries from its cache as shown in Figure 5D.

In certain situations, the operating system may be updating specific information that may not affect the whole fast-forwarding cache. In such a
20 situation, the operating system can instruct the network card to flush only the portion of its fast-forwarding cache relating to the updated portion of the routing information. For example, the operating system may maintain a mapping of IP addresses to Ethernet addresses. If a packet is not sent to a certain Ethernet address for a certain time period, the operating system may delete the mappings
25 relating to that Ethernet address and instruct the network card to flush only portions of the fast-forwarding cache relating to that Ethernet address. In this way, the network card can continue fast-forwarding packets based on the portions of the cache not flushed.

Figures 6A-6E depict the relevant fields of a number of different types of packets. Figure 6A depicts the fields that are used in fast-forwarding from an IP packet header 38. At offset 12 within the IP packet header 38 is a source address field 40 that holds an address for the source from which the packet originated. At offset 16 is a destination address field 42 that holds an address for the destination to which the packet is destined. The protocol field 44 is at offset 9 within the IP packet header 38 and indicates a protocol, such as TCP. Figure 6B shows fields of a TCP packet header 44. A source port field 46 is found at offset 0 and holds a value that identifies the source port of the application that sent the packet. A destination port field 48 is found at offset 2 and holds a value that identifies the destination port of the application to which the packet is sent. Figure 6C depicts a portion of a UDP packet header 50 that includes a source port field 52 at offset 0 and a destination port field 54 at offset 2. Figure 6D depicts an ICMP packet 56 that holds a type field 58 at offset 0 and a code field 60 at offset 1. Lastly, Figure 6E depicts an IPX packet header 62 that has a destination address field 66 at offset, source address field 68 at offset 18, and packet field 64 at offset 5.

Sometimes a packet itself may be subdivided into packet fragments, that are transmitted individually. When the packet fragments are received, they can be reassembled into the packet. However, packet fragments do not contain all the information of the packet identifier that is used to uniquely identify information in the fast-forwarding cache. Rather, a packet fragment contains a fragmented flag and an identifier field. When a packet is fragmented, a unique packet identifier is placed in each packet fragment and the fragmented flag is set. When a fragmented packet is received, the network card first checks if the identifier is present in a fragment cache. If it is the network card fast-forwards the packet, else it passes the fragment to the operating system and adds

it to the fragment cache when it is transmitted. In this way, the network card can fast-forward packet fragments.

Figures 7-10 are flow diagrams illustrating the processing performed by the network card and the operating system when processing packets. Figure 7 is a flow diagram of the routine of the network card that receives a packet from the network. This routine receives packets and determines whether fast-forwarding information for the packet is in the fast-forwarding cache. If fast-forwarding information is in the cache, the routine fast-forwards the packet, otherwise the routine sends the packet to the NDIS driver.

10 In step 701, the routine extracts the packet identifier from the received network packet. In step 702, if a row in the fast-forwarding cache matches the extracted packet identifier, then the routine continues at step 704, else the routine continues at step 703. In step 703, the routine stores the packet in the incoming cache and forwards the packet to the NDIS driver which forwards the packet to the operating system and completes. In steps 704-706, the routine fast-forwards the packet. In step 705, the routine updates the network source and destination address in the packet with the information from the fast-forwarding cache. In step 706, the routine adjusts the framing of the packet, if necessary. In step 707, the routine transmits the updated package to the NIC port indicated in the cache and completes. In addition, the routine updates the time-to-live field in the packet. If the value of the time-to-live field drops to below zero, then the routine drops the packet. The routine also updates the checksum of the packet as appropriate. For example, the routine recomputes the IP header checksum based on the updated value of the time-to-live field. If any option fields in the packet are set, then the routine sends the packet to the operating system rather than fast-forwarding the packet. The options fields indicate that further processing is needed before the packet can be forwarded. If for any reason the network card cannot process the packet, it sends the packet to the operating system for further

25

processing. For example, if the IP header checksum is not correct or the IP header time-to-live is zero, the network card forwards the packet to the operating system.

Figure 8 is a flow diagram of a routine for the network card to receive a packet from the NDIS driver. When the routine receives a packet from the NDIS driver, it updates the fast-forwarding cache accordingly and forwards the packet. In step 801, the routine creates a row and adds the row to the fast-forwarding cache. In step 802, the routine retrieves the packet from the incoming packet cache and adjusts the framing of the packet as necessary. In step 803, the routine transmits the packet to the NIC port indicated by the row and completes.

Figure 9 is a flow diagram of a routine of the operating systems to process a packet that is received from the NDIS driver. This routine checks its various filtering criteria to determine whether to drop a packet. If the packet is to be dropped, then the routine suppresses the forwarding of the packet to the network card. In step 901, the routine applies the various filtering criteria to the received packet. The filtering criteria can include dropping packets that are sent to a certain IP destination address or dropping packets that are directed to a certain TCP destination port. The filtering information is typically maintained by a network administrator. In step 902, if the received packet has been filtered out, then the routine completes, else the routine continues at step 903. By completing when the packet is filtered out, the routine effects the dropping of the packet. In step 903, the routine retrieves the new network source and destination addresses for the packet from the routing tables. In step 905, the routine sends the packet to the NDIS driver which forwards the packet to the network card. The routine is then done.

Figure 10 is a flow diagram of an operating system routine to update the routing tables. The routine receives new routing instructions, updates

its routing tables, and notifies the NDIS driver to flush the fast-forwarding cache. The NDIS driver in turn notifies the network card to flush its fast-forwarding cache.

While the present invention has been described with reference to a preferred embodiment thereof, those skilled in the art will appreciate that various changes of form and detail may be made without departing from the intended scope of the present invention as defined in the appended claims. For example, the fast-forwarding techniques of the present invention can alternatively be implemented by the NDIS driver rather than by the network card. With such an implementation, standard network cards can be used without modification. The NDIS driver would maintain the fast-forwarding cache and control the fast-forwarding of packets. When the NDIS driver receives a packet from a network card, it would check its cache to determine whether the packet should be fast-forwarded. If the packet is to be fast-forwarded, the NDIS driver would direct the appropriate network card to forward the packet. Otherwise, the NDIS driver sends the packet to the operating system. Also, although the techniques of the invention are described as being implemented as software executing on a network card with a processor, the techniques can be hardwired using logic circuits rather than software. Also, different fields from the packets may be selected to form a packet identifier for packets that are related, that is, to be fast-forwarded in the same way.

CLAIMS

We claim:

- 1 1. A method in a computer system for forwarding packets of
2 data from a source network to destination network, each packet being part of a
3 message, the computer system having a main central processing unit connected
4 via a bus to a network card, the network card being connected to the source and
5 destination networks, the network card having a cache memory, the computer
6 system having an operating system with instructions that are executed by main
7 central processing unit, the method comprising:
8 under control of the network card,
9 receiving a packet from the source network;
10 determining whether the cache memory contains routing
11 information for the packet;
12 when the cache memory contains such routing information,
13 transmitting the received packet on the destination network in accordance with
14 the routing information; and
15 when the cache memory does not contain such routing
16 information, sending the received packet to the operating system;
17 under control of the operating system,
18 receiving the sent packet;
19 determining whether the received packet should be dropped
20 based on filtering information; and
21 when the received packet should not be dropped,
22 retrieving routing information that indicates that the
23 receive packet should be transmitted on the destination network; and
24 sending the routing information to the network card;
25 and

26 under control of the network card,
27 receiving the sent routing information;
28 storing the received routing information in the memory
29 cache so that subsequent packets can be transmitted onto the destination network
30 without sending the subsequent packets to the operating system; and
31 transmitted the received packet on the destination network in
32 accordance with the routing information
33 wherein packets with the same packet identifiers can be forwarded on the
34 destination network without processing by the operating system, while packets
35 with other packet identifiers can be dropped based on the filtering criteria
36 processing of the operating system.

1 2. The method of claim 1 including:
2 under control of the operating system,
3 receiving an update to the routing information; and
4 in response to receiving the update, notifying the network
5 card to flush its cache memory.

1 3. The method of claim 1 wherein the computer system
2 includes a plurality of network cards and the cache memory is accessible by each
3 network card without interrupting the main central processing unit.

1 4. A method in a general-purpose computer system of
2 forwarding a network packet from a source network to destination network, the
3 computer system having a main central processing unit and a network card, the
4 network card being connected to the source and destination networks, the method
5 comprising:

6 receiving the network packet at the network card, the network
7 packet having a destination address;

8 analyzing the received network packet to determine whether the
9 network packet should be forwarded to its destination by the network card or
10 forwarded to the main central processing unit;

11 when it is determined that the network packet should be forwarded
12 to its destination by the network card, transmitting the network packet to the
13 destination network by the network card without intervention by the main central
14 processing unit; and

15 when it is determined that the network packet should be forwarded
16 to the main central processing unit, providing the received network packet to the
17 main central processing unit for further processing.

1 5. The method of claim 4 wherein the network card is a
2 multiport network card and wherein the network packet is received at a first port
3 and transmitted through a second port.

1 6. The method of claim 4 wherein the computer system further
2 comprises a bus through which the main control processing unit and network
3 card communicate and wherein the transmitting of the network packet is
4 accomplished without sending the network packet over the bus.

1 7. The method of claim 4, further comprising maintaining a
2 fast-forwarding cache on the network card for use when analyzing, the fast-
3 forward cache having routing information which is accessed to determine which
4 network packets are to be transmitted by the network card without intervention
5 by the main central processing unit.

1 8. The method of claim 4 wherein the main central processing
2 unit applies filtering to the network packet to determine whether to accept or
3 drop the network packet and wherein such filtering is enabled while the
4 transmitting by the network card without intervention by the main processing
5 unit is also enabled.

1 9. In a computer system having a main central processing unit
2 and a network card for interfacing the computer system with multiple networks, a
3 method of processing network packets that are received at the network card,
4 comprising:

5 having fast-forwarding active so that selected network packets that
6 are received at the network card are fast-forwarded to at least one destination
7 network by the network card without main central processing intervention; and

8 concurrently having filtering active so that the main central
9 processing unit applies filtering to at least some of the network packets received
10 at the network card to determine whether to accept or drop the network packets.

1 10. The method of claim 9 wherein at least some of the network
2 packets are Internet protocol (IP) network packets.

1 11. The method of claim 9 wherein the main central processing
2 unit provides the network card with routing information for fast-forwarding the
3 selected network packets.

1 12. The method of claim 9 wherein the network card provides to
2 the main central processing unit those network packets that are not to be fast-
3 forwarded.

1 13. A computer-readable medium containing instructions for
2 causing a general-purpose computer system to forward a network packet from a
3 source network to destination network, the computer system having a main
4 central processing unit and a network card, the network card being connected to
5 the source and destination networks by:

6 receiving the network packet at the network card, the network
7 packet having a destination address;

8 analyzing the received network packet to determine whether the
9 network packet should be forwarded to its destination by the network card or
10 forwarded to the main central processing unit;

11 when it is determined that the network packet should be forwarded
12 to its destination by the network card, transmitting the network packet to the
13 destination network by the network card without intervention by the main central
14 processing unit; and

15 when it is determined that the network packet should be forwarded
16 to the main central processing unit, providing the received network packet to the
17 main central processing unit to determine whether the network packet should be
18 dropped or forwarded to the destination.

1 14. The computer-readable medium of claim 13 wherein the
2 network card is a multiport network card and wherein the network packet is
3 received at a first port and transmitted through a second port.

1 15. The computer-readable medium of claim 13 wherein the
2 computer system further comprises a bus through which the main control
3 processing unit and network card communicate and wherein the transmitting of

4 the network packet is accomplished without sending the network packet over the
5 bus.

1 16. The computer-readable medium of claim 13, further
2 comprising maintaining a fast-forwarding cache on the network card for use
3 when analyzing, the fast-forward cache having routing information which is
4 accessed to determine which network packets are to be transmitted by the
5 network card without intervention by the main central processing unit.

1 17. The computer-readable medium of claim 13 wherein the
2 main central processing unit applies filtering to the network packet to determine
3 whether to accept or drop the network packet and wherein such filtering while
4 the transmitting by the network card without intervention by the main processing
5 unit is enabled.

1 18. A computer system having a main central processing unit
2 and a network card for interfacing the computer system with multiple networks,
3 the computer system for processing network packets that are received at the
4 network card in which fast-forwarding is active so that selected network packets
5 that are received at the network card are fast-forwarded to at least one destination
6 network by the network card without main central processing intervention and in
7 which filtering is concurrently active so that the main central processing unit
8 applies filtering to at least some of the network packets received at the network
9 card to determine whether to accept or drop the network packets.

1 19. The system of claim 18 wherein at least some of the network
2 packets are Internet protocol (IP) network packets.

1 20. The system of claim 18 wherein the main central processor
2 provides the network card with routing information for fast-forwarding the
3 selected network packets.

1 21. The system of claim 18 wherein the network card provides
2 to the main central processing unit, those network packets that are not to be fast-
3 forwarded.

1 22. A method in a general-purpose computer system of
2 forwarding a network packet from a source network to destination network, the
3 computer system having an operating system, a network driver, and a network
4 card, the network card being connected to the source and destination networks,
5 the method comprising:

6 under control of the network card,
7 receiving the network packet having a destination address
8 from the source network; and

9 forwarding the received network packet to the network
10 driver;

11 under control of the network driver,
12 analyzing the received network packet to determine whether
13 the network packet should be forwarded to its destination by the network card or
14 provided to the operating system;

15 when it is determined that the network packet should be
16 forwarded to its destination by the network card, instructing the network card to
17 transmit the network packet to the destination network; and

18 when it is determined that the network packet should be
19 provided to the operating system, providing the received network packet to the
20 operating system for further processing; and
21 under control of the network card,
22 when instructed to transmit the received packet, transmitting
23 the network packet to the destination network
24 so that a decision to forward a network packet is made by the
25 network driver.

1 23. The method of claim 22 wherein computer system has a
2 plurality of network cards and wherein the network packet is received at a first
3 network card and transmitted through a second network card.

1 24. The method of claim 22, further comprising maintaining a
2 fast-forwarding cache by the network driver for use when analyzing, the fast-
3 forward cache having routing information which is accessed to determine which
4 network packets are to be transmitted by the network card without intervention
5 from the operating system.

1 25. The method of claim 22 wherein the operating system
2 applies filtering to the network packet to determine whether to accept or drop the
3 network packet and wherein such filtering is enabled while the network packets
4 are fast-forwarded by the network driver without intervention by the operating
5 system.

1 26. The method of claim 22 wherein the network driver is an
2 NDIS driver.

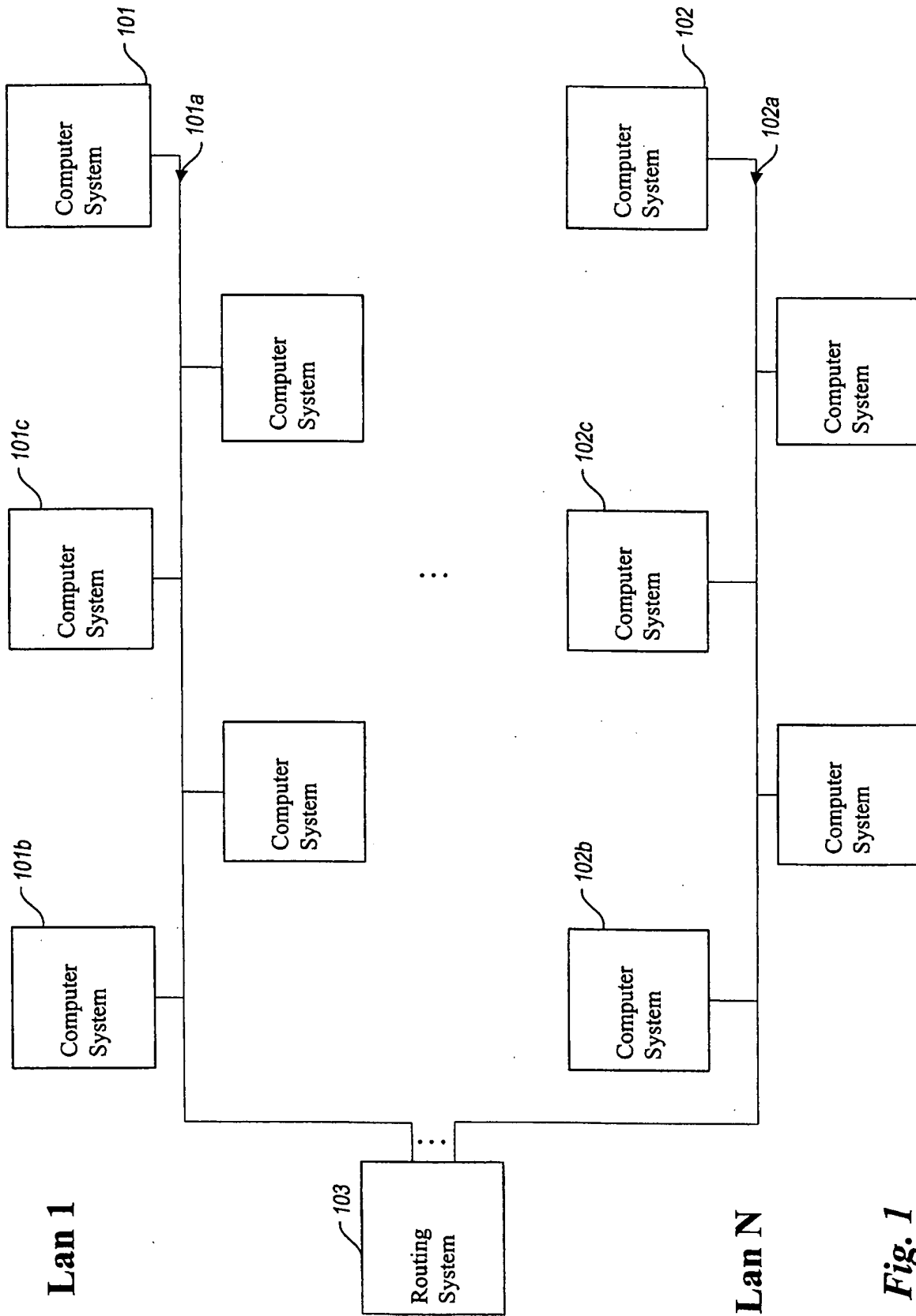


Fig. 1

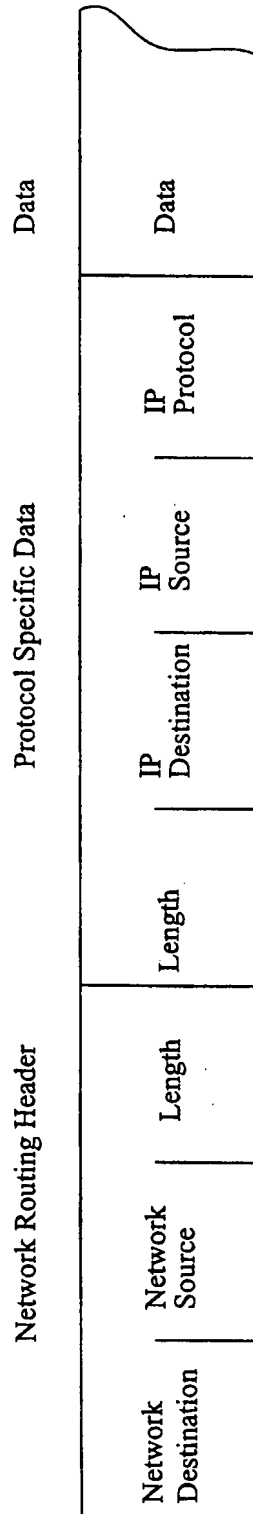


Fig. 2

3/11

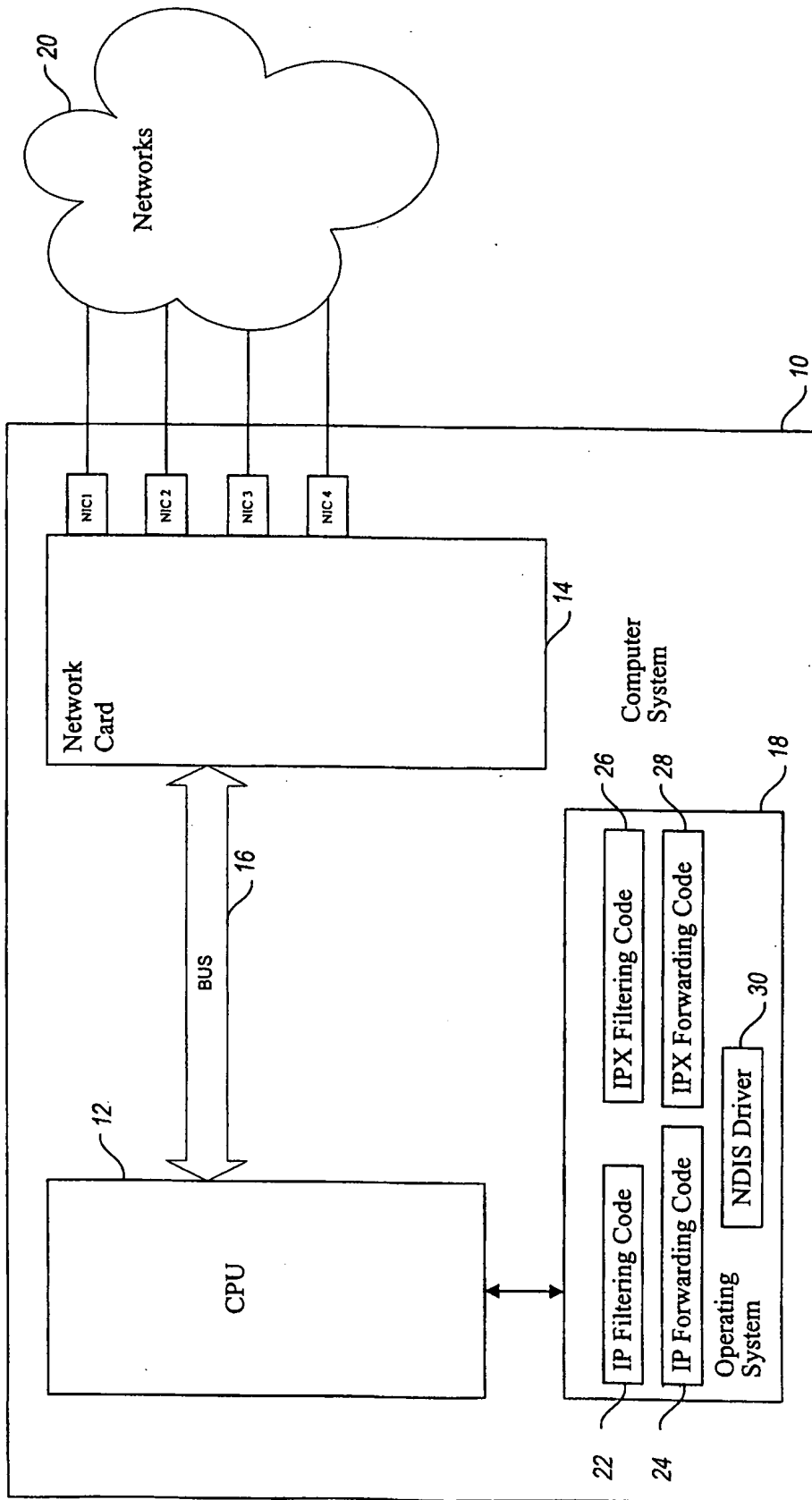


Fig. 3

4/11

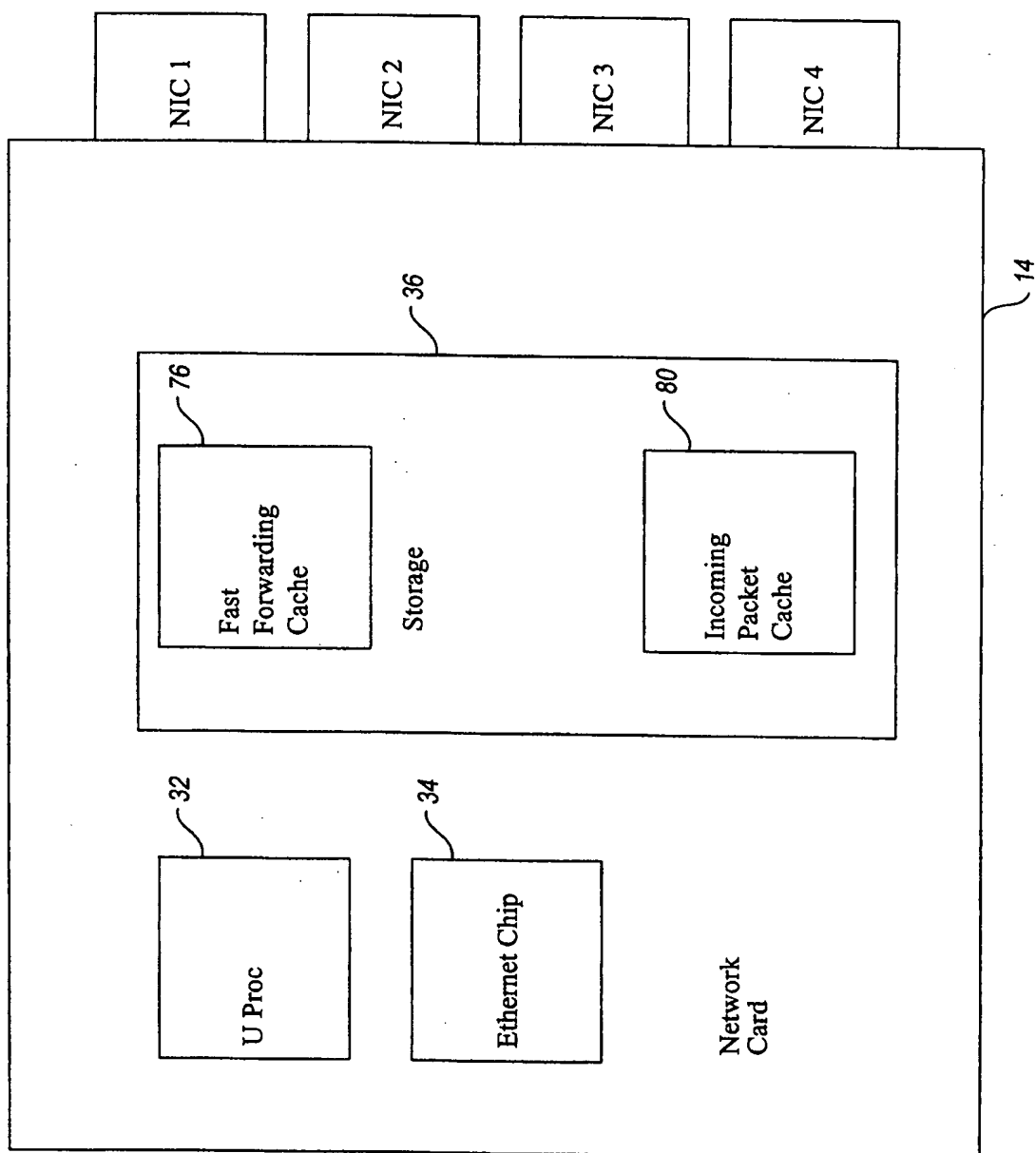


Fig. 4

5/11

Fast-Forwarding Cache**Fig. 5A**

Packet Identifier	Network Addresses	Framing	NIC

Fig. 5B

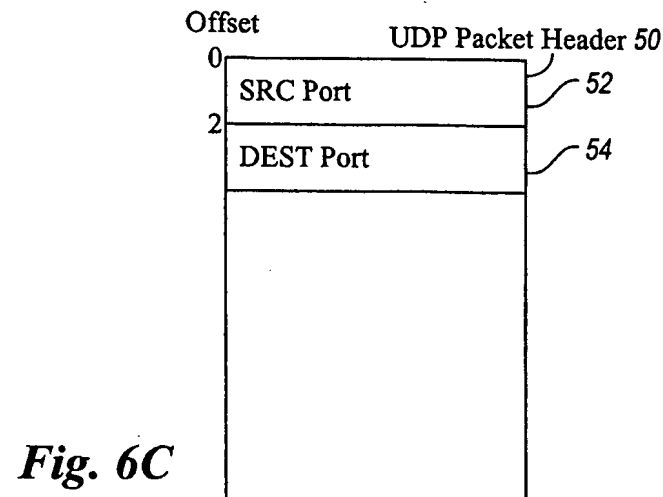
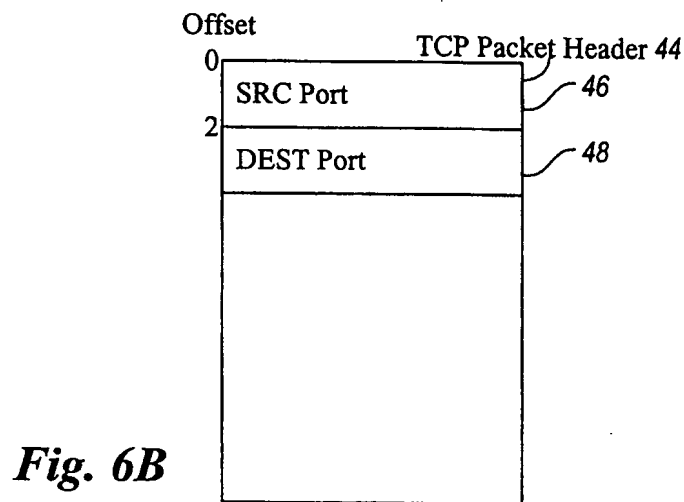
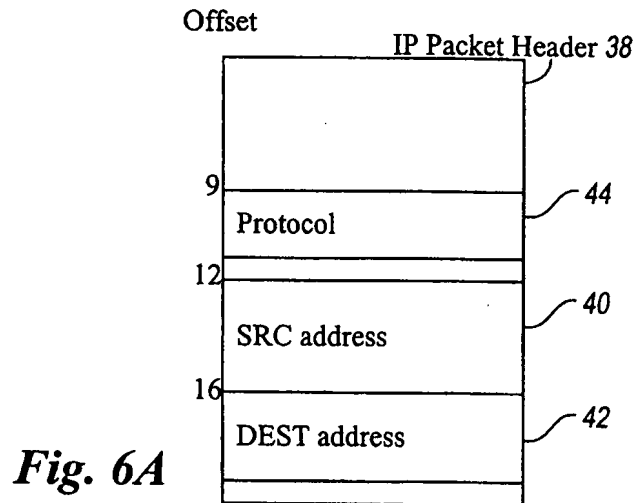
<A5, B2, TCP, 14, 25>	B2, E7	802.3	3

Fig. 5C

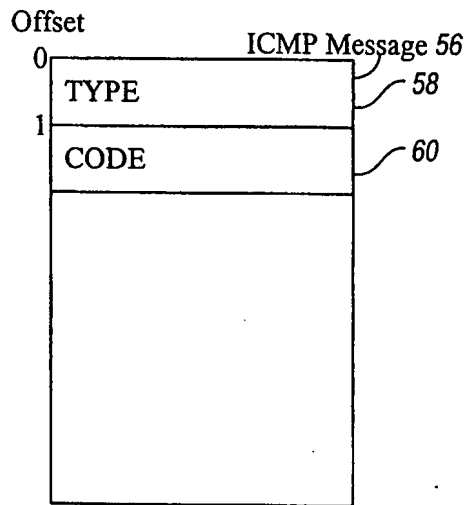
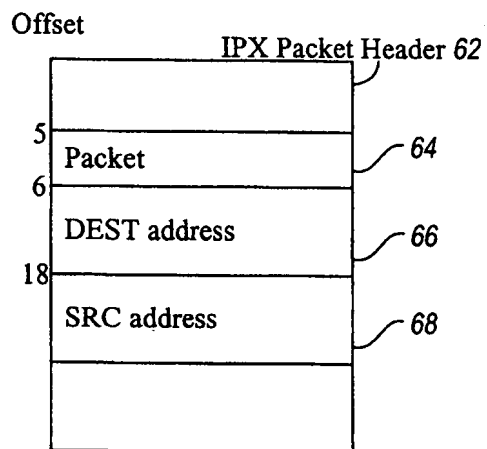
<A5, B2, TCP, 14, 25>	B2, E7	802.3	3
<E1, B2, UDP, 36, 97>	B3, A5	Ethernet	1

Fig. 5D

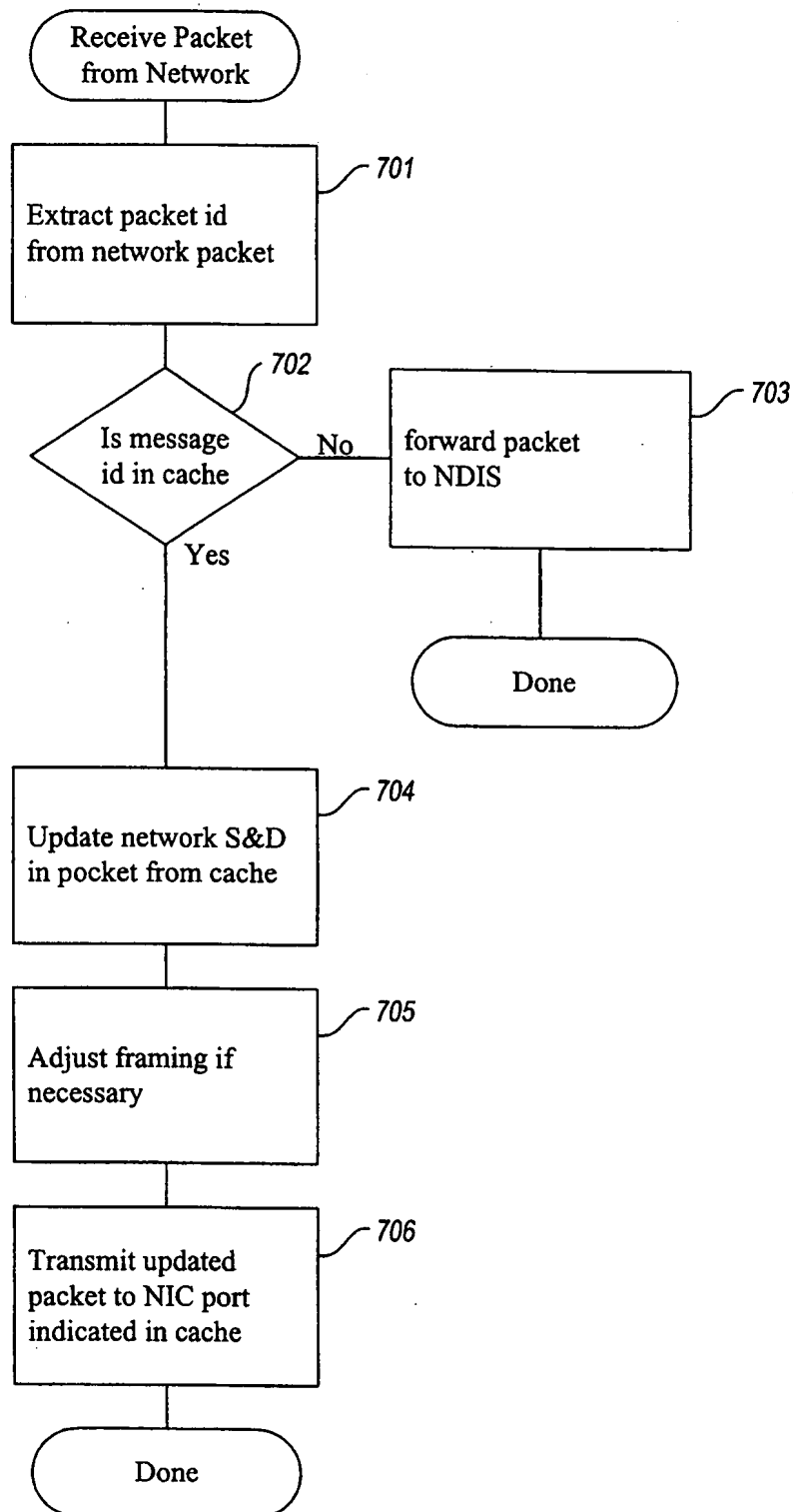
6/11

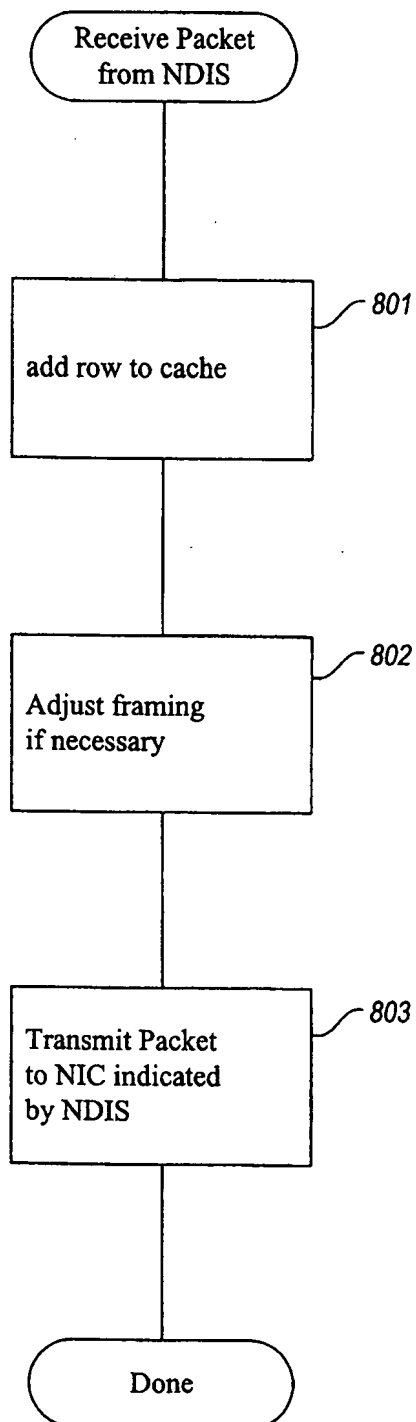


7/11

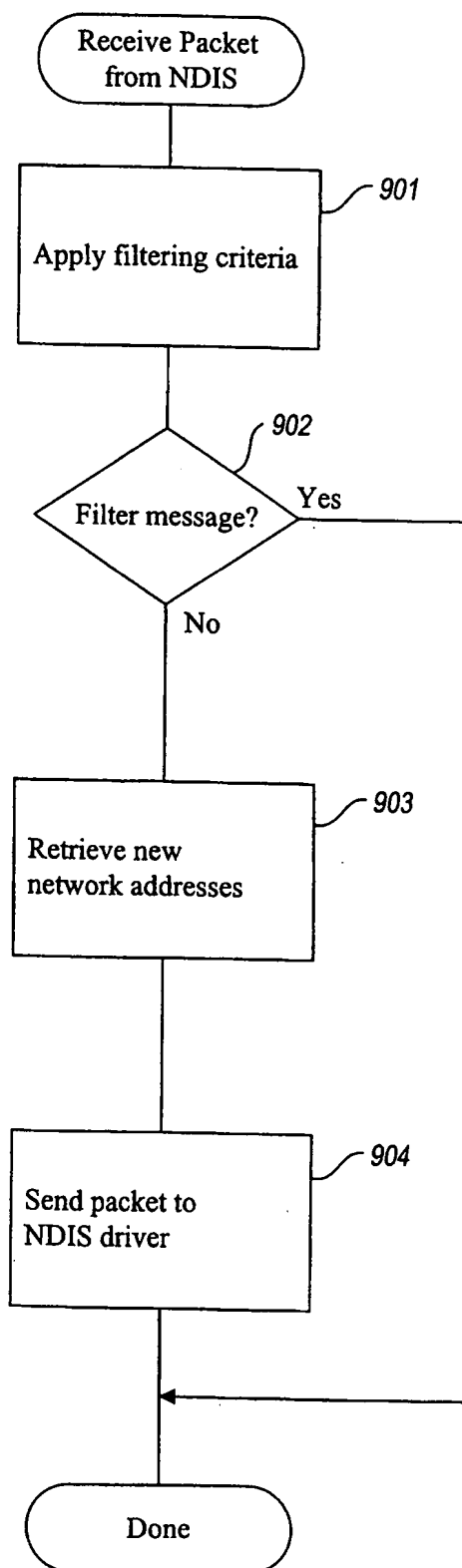
**Fig. 6D****Fig. 6E**

8/11

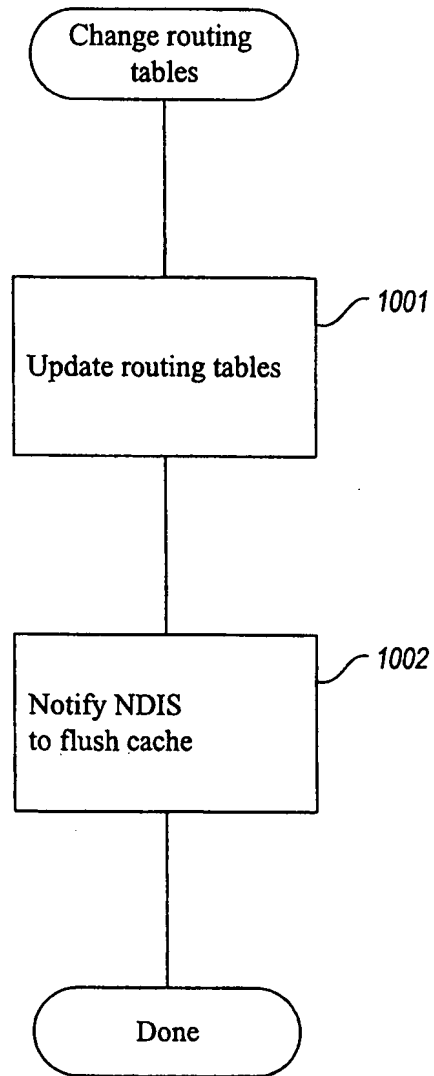
*Fig. 7*

9/11***Fig. 8***

10/11

*Fig. 9*

11/11

*Fig. 10*